Evaluating Location Dependent Queries Using ISLANDS

Marie Thilliez, Thierry Delot

LAMIH Laboratory - CNRS UMR 8530
University of Valenciennes
Le Mont Houy
59313 Valenciennes Cedex 9 France
{firstname.lastname}@univ-valenciennes.fr

Abstract. The recent emergence of handheld devices and wireless networks implies an exponential increase in the numbers of terminals users. Given this increase, today's service providers have to propose new applications adapted to mobile environments. In this article, we focus on distributed proximity M-services, in which several handheld devices, situated in close physical proximity to one another, can communicate and exchange data. Proximity M-services exploit a combination of mobile devices and heterogeneous mobile and/or fixed networks, and require a high degree of flexibility in order to permit easy and rapid application development. Because these applications are based on the Hybrid Peer-To-Peer (P2P) software architecture, such problems as scalability, deployment, security, reliability and information retrieval in M-services, can be more easily resolved than in other architectures. Within the framework of this software architecture, we focus on the localization problematic. Existing localization solutions are not well adapted to the mobility, dynamicity and heterogeneity of the Proximity M-service environment. Our solution to this lack of adaptation is ISLANDS, a service designed to aid the user in identifying and locating information and/or services within a communication area. ISLANDS is a decentralized service derived from the directory service model and adapted to the management of numerous distributed resources. In addition, ISLANDS provides querying facilities and a query evaluation model adapted to distributed directory management. Since most ISLANDS users are mobile, location dependent queries are also supported, thus enabling users to track down the closest theater or, more importantly in an emergency, the closest general practitioner's office. Doing so obviously requires pinpointing the location of the query issuer, and one of the original aspects of our approach is the adaptability of the localization process : given that few handheld devices today are equipped with geo-localization features, ISLANDS is able to return approximate solutions by estimating the position of the client terminal, although ISLANDS can also use GPS-like techniques if available.

Keywords : Localization Service, Location Dependent Queries, Evaluation

1 Introduction

The emergence of both handheld devices and wireless networks [19] implies an exponential increase in the numbers of terminals users. Given this increase, today's service providers have to offer new proximity services adapted to mobile environments [1, 3]. Among these services, proximity applications [16] deployed in highly distributed environments permit new possibilities for users of these handheld devices. These proximity applications exploit communication areas formed dynamically by juxtaposing several wireless and mobile networks. For example, a communication area can result from the association of a wireless LAN (Local Area Network) and a wireless PAN (Personal Area Network). Wireless communication areas are also highly dynamic since they evolve in relation to user mobility. These areas allow communications between different users who are physically close to one another. Indeed, depending on the location of these users, a certain set of services is proposed. For example, using a proximity service, users could buy goods, exchange data or communicate with other users. In addition, the set of services can evolve as the user moves from one area to another. A proximity service is created spontaneously when several users form a communication area and want to share information. This service then evolves dynamically in relation to the movements of the participants and terminates when there are no longer any participants.

Due to the dynamicity and the heterogeneity of both networks and devices, a high degree of flexibility is required to exploit proximity services. In an earlier article [16], we demonstrated the importance of the hybrid Peer-To-Peer (P2P) architecture model [24] in basic proximity applications. Clearly, the partial centralization and the flexibility of the P2P model make proximity applications developed using this model much more adaptable to changing environments.

In such environments, the information is distributed through different devices. Due to user mobility, the information available in the communication area evolves rapidly, and localization services are needed to provide correct and up-to-date information to users. Without such mechanisms, users cannot exploit the proximity services since they are unable to access the available information. For example, in a proximity electronic commerce application, the potential client has to be able to retrieve the different vendors and their propositions in order to benefit from the offers. Because existing localization solutions (naming services, trading services, discovery services, …) do not deal well with the constraints imposed by proximity applications, in terms of distribution, dynamicity and heterogeneity of both terminals and networks, we propose ISLANDS[1] [18], a new distributed localization service, exploiting directory services technology and dedicated to mobile environments.

In a proximity applications, due to the users mobility, we have also to be able to evaluate queries in function of users locations. Thus, based on ISLANDS, we focus on the evaluation of location dependent queries such as "Where are the music stores closest to me ?". Location dependent queries are evaluated according to the user's physical location, and responses are designed to help the user localize the resources

[1] **I**nformation and **S**ervices **L**ocaliz**A**tio**N** and **D**iscovery **S**ervice

needed easily and precisely. The process for locating the user proposed in the paper and required for the evaluation of location dependent queries does not rely only on existing localization techniques, such as GPS, since they are only available on few handheld terminals today. Indeed, using ISLANDS, user location can also be estimated using the localization data usually stored in directory services for computing approximate query results.

The rest of this paper is organized as follows : Section 2 gives an overview of the localization service ISLANDS and present existing solutions in location queries evaluation and their limitations. Our proposition to evaluate location dependent queries is explained in section 3, with special attention given to the procedure for estimating the location of a mobile user. And finally, in section 4, in conclusion, we present our prototype and present the perspectives for future research.

2 ISLANDS

2.1 An overview

A participant in a proximity service has to be able to locate the data available in his/her communication area, not only the information stored on the participant's device but also the information managed by the system's remote peers. Since existing localization solutions present severe drawbacks, we have proposed a dedicated solution: ISLANDS, an Information and Services LocalizAtioN and Discovery Service [18]. ISLANDS is based on the directory services technology because, compared to both naming and trading services, this technology provides very interesting features such as scalability, querying facilities, authentication and so on. However, the current directory services present also severe limitations in our environment. Thus, in ISLANDS, we have extended the directory services technology to be adapted to the constraints imposed by mobile, heterogeneous and dynamic environments. In order to support these constraints and to exploit the benefits of the underlying hybrid P2P architecture, our solution does not rely on a centralized server but rather on the deployment of ISLANDS for each peer able to interact in the communication area. As shown in Fig. 1, based on the hybrid P2P Model, two types of peers are distinguished : the light peers and the central peers. Peer type essentially depends on the underlying hardware configuration. Central peers generally correspond to robust servers, whereas light peers correspond to handheld devices. Central peers, as their name implies, centralize information to share it with the other peers. The most of time, every central peers are selected at the application deployment. Light peers correspond to thin clients exploited by users to submit their queries and consult the results. We choose to deploy ISLANDS on every peers (central or light) in order to stand up to the inherent mobility and dynamicity of the system. Naturally, ISLANDS can be customized in terms of the peer’s resources (CPU power, memory space …). For example, a powerful peer (as is generally the case for central peers) can easily store and share information about the other peers. On the other hand, the localization service as deployed for a light peer may only be able to store a small amount of information locally in addition to providing users with a means for retrieving the information stored on remote peers. Thus, the distribution between the instances of ISLANDS services is completely transparent for users. Indeed, in proximity applications, the data and services set is distributed over the different peers present in the communication area. ISLANDS queries are evaluated throughout the entire communication area and thus, if necessary, are forwarded automatically to remote directory services.

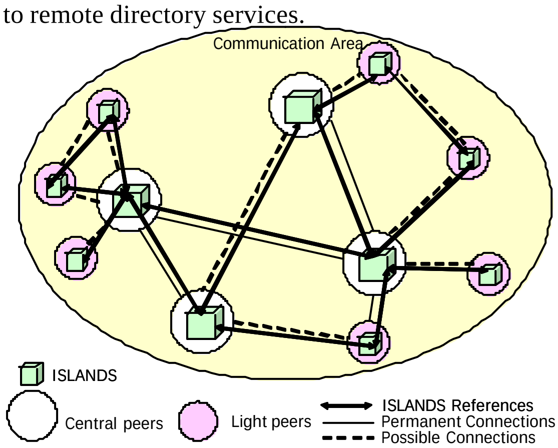

Fig. 1. ISLANDS Distribution

2.2 Location Queries

In the mobile context considered in this article, the different participants have to be able to query information in relation to their location. For example, a participant may want to identify and retrieve the location of the closest general practitioner. Traditionally, directory services propose very simple query languages. However, this is very restrictive. With ISLANDS, in order to select information in terms of location or proximity, several simple and user-friendly operators are used to verify proximity constraints : inside, closest and close [17]. First, the inside operator is used to retrieve elements within one area. For example, it can be used to retrieve every vendors at the ground floor of a shopping mall. Then, the closest operator is used to access a particular element located the shortest distance from the query issuer or from a specified location parameter. It can be used, for example, to retrieve the fast food closest to me or closest to Virgin. And finally, the close operator is an evolution of the closest operator. This operator is used to retrieve several elements close to the issuer or close to a specified location parameter. It may also be used with a distance parameter. This optional parameter is an integer representing the number of meters, which defines the maximal distance between the specified target and the issuer (or a specified location).

Since the presentation of the concept in 1992 [10], querying location dependent information in mobile environments has become an important field of research. A lot of studies concern questions of data management in mobile devices and the information about devices' location [4, 12]. Of those, several are worth mentioning. Sisla et al. [14, 15] have proposed the Moving Object Spatio-Temporal (MOST) data model to manage the locations of moving objects and to predict their future locations. The DOMINO project [20, 21, 22, 23], which complements our work quite nicely, proposes a model that represents moving objects in a database and tracks them efficiently. These same authors also propose Future Temporal Logic (FTL) as a query language for such a model. Several other interesting works deal with processing continuous location-dependent queries in mobile environments [5, 7, 8, 11]. These studies concentrate on query processing on mobile terminals. Such processing relies on a centralized environment and the use of handheld devices equipped with localization techniques, such as GPS. In this article, we have focused on evaluating issuer location in order to improve the quality of location-dependent query results. Similar studies have also been completed using URLs in a web context [2], but to the best of our knowledge, the problems of mobility and dynamicity have not yet been dealt with prior to the work described in this article.

3 Evaluating Location Dependent Queries

One of the main difficulties of the query engine concerns the evaluation of location-based queries. First, the evaluator has to define whether the query is location-aware or location-dependent [13]. A location-aware query, such as “retrieve the fast foods close to Virgin”, does not depend on the issuer's location. On the other hand, a location-dependent query, such as “retrieve the TV set vendor closest to me”, requires that the user's position be known. In this section, we firstly focus on the evaluation of the client location. Then, we present the evaluation of the location queries.

3.1 Evaluation of the Location

One of the original aspects of our approach is the adaptability of its localization process, which can be adapted to the resources of the underlying peer. The process can be based on geographic localization technologies like GPS; however, given that today's handheld devices often do not have GPS-like features, localization will generally be based on the location data stored in the Directory Information Tree (DIT), particularly the referenced directory services. Using stored data means that localization can be approximated. Some studies [9, 25] have proposed broadcasting geographic information to wireless device users so that they can evaluate their position. However, such solutions make it difficult for a particular user to estimate the correctness and the precision of the broadcast information. To increase the precision of the location information broadcast to users, one solution is to increase the number of servers used to broadcast location information. Unfortunately, seeing as the information may come in from several different servers, this solution complicates matters by making it very difficult for the handheld devices to choose which location information should be used. Our solution, which exploits the P2P environment, relies on information-sharing between neighboring peers, as explained below.

If the query issuer is using a stationary computer, its location is well known, and the location evaluation is negligible. Indeed, stationary computers always have the same location, which is specified only once. On the other hand, defining the location of mobile peers is a major difficulty because issuer location can change at any moment and potentially must be re-evaluated for each evaluated query. For those cases when location cannot be provided by geo-localization techniques like GPS, we propose to compute approximate and relative locations that are nonetheless sufficient for evaluating location dependent queries. Specifically, the approximate location of one peer is determined using location information from the peers connected to it.

```
<locationDescription>
  <symbolicLocation>
    <building name = « Building 1 »>
      <floor name = « first »>
         <section name = « North »>
            <room name = « Virgin »>
            </room>
         </section>
      </floor>
    </building>
 </symbolicLocation>
 <physicalPosition>
      <lat> 27.7 </lat>
      <long> -15.1 </long>
      <alt> 197.4 </alt>
 </physicalPosition>
</locationDescription>
```

Fig. 2. Example of a locationDescription attribute file

Indeed, the location evaluation is based on remote peers data. This information can be stored locally on the peer. Indeed, for each remote neighbored peer, several attributes are retrieved and can be stored on the local peer. To estimate the issuer location, we use four of these attributes for each remote peer :

- The *locationDescription* attribute describes the geographic location of the referenced directory service. As shown in Fig. 2, the symbolic location (building, room …) and/or the physical location (using GPS for example) [6] can be represented in XML because it allows a trace of the location data semantic to be preserved and also facilitates location data management.
- The *locationLastUpdate* attribute represents the date of the last update of the *locationDescription* attribute file.
- The *connectionRange* attribute provides a range which allows to estimate the distance between the peer and the remote peer.
- The *connectionState* attribute allows to know if the peer is connected to the remote peer.

The algorithm (the evaluation function) described in Fig. 3, used to evaluate the location of a light peer employs an approximation degree, calculated from the *locationLastUpdate* and the *connectionRange* attributes, to estimate the quality of the computed location. The smaller the degree, the better location. This algorithm provides a list of couples (location, approximation degree), sorted by approximation degree. In this algorithm, the *Nb* variable is defined in terms of environmental constraints and indicates the maximal number of locations to be computed. The smaller the *Nb* variable is, the shorter the execution time of the evaluation will be.

```
Function Evaluation (Peer Directory Service) return LocationList
{
Nb := LocationListNb() ; //returns the number of desired elements in the LocationList
RefPeerEntriesList := RefPeerEntriesList() ; //retrieves the set of referenced peers
ConnectedPeerList := Connected(RefPeerEntriesList); //the attribute connectionState >
0
nbConnectedPeers := nbPeer(connectedPeerList) ;
LocationList := null ;
If ConnectedPeerList is not null then
    {
     Int d[nbConnectedPeers] ;
     Int r[nbConnectedPeers] ;
     For i=1 to nbConnectedPeers loop
  {
       d[i] := coeff_d(connectedPeerList, i) ; //defines the d coefficient : the
difference between the current date and the locationLastUpdate attribute
      r[i] := coeff_r(connectedPeerList, i) ; //defines the r coefficient in function
of the connectionRange attribute
    approxDegree := approxDegree(d[i], r[i]) ; //returns a*r[i] if the peer i is a
central peer or b*d[i] + c*r[i] if the peer i is a light peer
    locationList := concat(locationList, (location(i), approxDegree(i))) ;
  }
If Nb > nbConnectedPeers then Nb := nbConnectedPeers ;
LocationList := Sort(LocationList, Nb); //returns a list of Nb couples sorted in
function of the approximation degree
Return LocationList ;
}
```

Fig. 3. Evaluation Function

First, if the light peer is connected to other peers, the d_i and r_i variables for each connected referenced peers are defined. The *d* variable corresponds to the difference between the current date and the *LocationLastUpdate* attribute. Thus, the larger *d* is, the more out-of-date the location data is. The *r* variable defines a proximity index corresponding to the *connectionRange* attribute (which defines an approximate value of the distance separating the light peer from their connected peers in function of the network rate quality) in order to select the closest peers. The d_i and r_i variables are stored in two tables : respectively tables *d* and *r*. In the case of central peers, the d_i variable is null because their location is always updated. Then, for each connected peer, we calculate their approximation degree. This degree is $a*r_i$ for a central peer or $b*d_i + c*r_i$ for a light peer. The *a*, *b* and *c* coefficients allow the priorities used to compute the "best" locations to be balanced. Then, a location list of *Nb* couples is sorted in function of this approximation degree. *Nb* corresponds either to the *Nb* variable defined in the evaluation function or to the number of connected peers (*nbConnectedPeers*) if *Nb*>*nbConnectedPeers*.

Example :

In this example, a peer *p* has express a location dependent query, thus, to evaluate this query, his/her location has to be evaluated. The following illustration of the location evaluation is based on a configuration in which the peer *p* has 6 neighbored peers : 3 central peers and 3 light peers. The different attributes of the neighbored peers are stored on the peer *p*, these attributes are described in the table 1.

Table 1. Example of referenced directory services entries

Referenced Light Peers	*Referenced Central Peers*
LP = LP1 locationDescription = «LP1.xml » locationLastUpdate = Oct-14th-2003 1:00 pm connectionRange = 2 connectionState = 0 …	CP = CP1 locationDescription = «CP1.xml » locationLastUpdate = sysdate connectionRange = 60 connectionState = 30 …
LP = LP2 locationDescription = «LP2.xml » locationLastUpdate = Oct-14th-2003 2:33 pm connectionRange = 3 connectionState = 30 …	CP = CP2 locationDescription = «CP2.xml » locationLastUpdate = sysdate connectionRange = 90 connectionState = 0 …
LP = LP3 locationDescription = «LP3.xml » locationLastUpdate = Oct-14th-2003 2:16 pm connectionRange = 30 connectionState = 40 …	CP = CP3 locationDescription = «CP3.xml » locationLastUpdate = sysdate connectionRange = 30 connectionState = 60 …

The current date in the example is Oct-14th-2003 at 2:43 pm. a, b and c are respectively set to 1, 10 and 1. *Nb* is set to 3. Then, the Evaluation function (Fig. 3) is executed.

```
Function Evaluation :

Nb := 3
The RefPeerEntriesList is {LP1, LP2, LP3, CP1, CP2, CP3}.
The ConnectedPeerList is {LP2, LP3, CP1, CP3}.
nbConnectedPeers := 4

The set of variables is defined : d = [10,27,0,0], r = [3,30,60,30]
For each connected peer, an approximation degree is calculated (a*r[i]
for a central peer or b*d[i] + c*r[i] for a light peer)
Thus, the locationList is :
 {(LP2.xml, 103), (LP3.xml, 300), (CP1.xml, 60),(CP3.xml, 30)}

Then, the list is sorted in function of approximation degrees.
Thus the final result is the LocationList = {(CP1.xml, 30), (CP3.xml,
60), (LP2.xml, 103)}
```

This location evaluation returns a list of locations sorted in terms of their approximation coefficients. The approximation coefficient is very important and will facilitate the location query evaluation described in the next section. In rare cases, the light peer location cannot be evaluated because the evaluation function returns a null/empty list. In this case, an interactive system can be proposed which will allow the user of the light peer to describe his/her location.

3.2 Evaluation of Location Queries

The previous section explained how the issuer location is determined when the query is location-dependent. Then, let us call QL the location resulting from the query and PL, the parameter location, which can be explicitly specified in the location operator or computed. For example, to retrieve the fast food closest to me, the QLs will be every fast food locations and the PL is my location (one XML file if my location is exact or several XML files if my location is computed using the location evaluation solution described in the precedent section). For each location operator, the evaluator must compare each PL of the locationList with each QL. If the query is location aware or if the peer location is fixed and known, there is only one element in the locationList and the approximation degree is null. For example, for the location-aware query : “Retrieve the fast foods close to Virgin ?”, each QL corresponds to each location of one fast food in the shopping mall and PL corresponds to the list containing the unique couple {(VirginLocation.xml, 0)}.

We choose to base the location description on the same fixed DTD to facilitate the comparison between XML files. Therefore, for each proximity application, a DTD could be defined. Based on fixed DTD, the comparison of the different location files is more easily computed. In a XML location file, the location can be represented in different ways : for example, such as the location described in Fig. 9, the location is described in symbolic data and in physical data. Indeed, some central peers can be provided with a GPS module.

Firstly, we consider the inside operator, which has a LocationType parameter. The comparison of two locations is possible only if the two location XML files contain a symbolic description. The LocationType parameter represents a XML element, for example the “stage” element to retrieve the fast foods of my stage. Then, each PL is compared with each QL : when there is a match to the LocationType element, the QL is selected. And finally, for each selected location, a correctness degree is computed : Correctness_Degree = 1/(Approximation_Degree + 1).

Table 2. Comparisons in function of XML files contents

		QL’s XML file content		
		Symbolic representation	Physical representation	Symbolic & physical representation
PL’s XML file content	Symbolic representation	Symbolic Comparison	-	Symbolic Comparison
	Physical representation	-	Physical Comparison	Physical Comparison
	Symbolic & physical representation	Symbolic Comparison	Physical Comparison	Physical Comparison

Secondly, let us consider the close and closest operators. As shown in table 2, the comparison between two locations is computed in different ways according to the XML files contents. For example, if the PL corresponds to symbolic data, the comparison is possible if the QL is represented in symbolic data or in both symbolic and physical data.

When there is a match between two locations, a similarity coefficient is computed. This coefficient is based on the percentage of values similarities in the case of symbolic representation comparison. In the case of physical representation, this similarity coefficient is based on the distance between the two physical locations : the larger the distance, the smaller the similarity coefficient. This distance is $v[(lat1-lat2)^2+(long1-long2)^2+M*(alt1-alt2)^2]$. The M constant is used to penalize the altitude. Then, a correspondence table is established in function of the application environment, this table allows to obtain a same similarity coefficient for the different representations. An example of this table is shown in the Table 3.

Table 3. An example of correspondence table for the similarity coefficient between two locations

Physical comparison (distance in meters)	Symbolic comparison (similarities in %)	Similarity Coefficient
<100 m	>0%	10%
<80 m	>20%	30%
<60 m	>40%	50%
<40 m	>60%	70%
<20 m	>80%	90%

Thus, for each comparison between a PL and a QL, we obtain a similarity coefficient.

If the location operator is close, when the distance parameter is specified, the query manager selects the QL solutions for which the similarity coefficient corresponds to a distance lower than the distance parameter. When the distance parameter is not specified, the query manager computes a correctness degree for each couple (PL,QL). The correctness degree is based on both the similarity coefficient and the approximation degree : Correctness_Degree = Similarity_Coefficient / (Approximation_Degree +1). Then, QL is sorted in function of a correctness degree. And the query manager selects a set of QLs, which have a correctness degree superior with a degree specified by the user. If the location operator is closest, the query manager selects the best QL solution in terms of the correctness degree.

4 Conclusion & Perspectives

In this paper, we have presented the evaluation of location dependent queries in ISLANDS. Indeed, ISLANDS is a localization service, designed to support mobility and thus, it offers a query evaluator that provides distribution transparency and supports location dependent queries. Indeed, to evaluate location-dependent queries, the originality of our approach resides in the possibility of using not only localization techniques like GPS but also the data description localization that is usually stored in directory services. A prototype of ISLANDS, as presented in this paper, has been implemented. The Proximity Electronic Commerce application is selected to validate the prototype. In this prototype, the different peers are connected with Bluetooth or

Wifi technologies. Pocket PCs Compaq Ipaq H5450 are used as light peers and represent the potential clients. Their data are stored in XML files. The central peers represent the vendors and their data are stored in the OpenLDAP directory server. The content of these directories is exported in DSML files in order to evaluate location dependent queries. To optimize the performances at the time of the updates, the different attributes of the neighbored peers are stored in indexes files.

In the near future, we will have to consider query distribution optimization. Indeed, for the moment, the query is forwarded to every connected peers; however, this may prove to penalize light peers (query execution time, resource consumption). Consequently, distribution strategies are needed for optimizing the query evaluation.

5 References

1. M. Bechler, H. Ritter, J. H. Schiller, Quality of Service in Mobile and Wireless Networks: The Need for Proactive and Adaptive Applications, Int. Conference on System Sciences (HICSS), 2000.
2. O. Buyukkokten, J. Cho, H. Garca-Molina, L. Gravano, N. Shivakumar, Exploiting geographical location information of web pages. Int. ACM SIGMOD Workshop on the Web and Databases (WebDB), 1999.
3. Y.-F. R. Chen, C. Petrie, Ubiquitous Mobile Computing, IEEE Internet Computing, Vol. 7, N° 2, 2003.
4. M. H. Dunham, V. Kumar, Location dependent data and its management in mobile databases, Int. Workshop on Mobility in Databases and Distributed Systems (MDDS), 1998.
5. G. Gok, O. Ulusoy, Transmission of Continuous Query Results in Mobile Computing Systems, Information Sciences, vol.125, no.1-4, 2000.
6. J. Hightower, G. Borriello, Location Systems for Ubiquitous Computing, IEEE Computer, 2001.
7. S. Ilarri, E. Mena, A. Illarramendi, Monitoring Continuous Location Queries using Mobile Agents, East-European Conf. on Advances in Databases and Information Systems (ADBIS), 2002.
8. S. Ilarri, E. Mena, A. Illarramendi, Dealing with Continuous Location-Dependent Queries: Just-in-Time Data Refreshment, IEEE Int. Conf. on Pervasive Computing and Communications (PerCom), 2003.
9. J. Il-dong, Y. Young-ho, L. Jong-hwan, K. Kyungsok, Broadcasting and Caching Policies for Location-Dependent Queries in Urban Area, ACM Int. workshop on Wireless sensor networks and applications (WMC), 2002.
10. T. Imielinski, B.R. Badrinath, Querying in Highly Mobile and Distributed Environments, Int. Conf. on Very Large DataBases (VLDB), 1992.
11. K. Lam, O. Ulusoy, T.S.H. Lee, E. Chan, G. Li, An Efficient Method for Generating Location Updates for Processing of Location-Dependent Continuous Queries, Int. Conf. on Database Systems for Advanced Applications (DASFAA'01), 2001.
12. D. L. Lee, J. Xu, B. Zheng, W-C. Lee, Data Management in Location-Dependent Information Services, IEEE Pervasive Computing, 2002.
13. A. Y. Seydim, M. H. Dunham, V. Kumar, Location Dependent Query Processing, Proceeding of MobiDE, 2001.
14. A. P. Sistla, O. Wolfson, S. Chamberlain, S. Dao, Modelling and Querying Moving Objects, Int. Conf. on Data Engineering (ICDE), 1997.

15. A. P. Sistla, O. Wolfson, S. Chamberlain, S. Dao, Querying the Uncertain Position of Moving Objects, Temporal Database: Research and Practice, LNCS, 1997.
16. M. Thilliez, T. Delot, S. Lecomte, N. Bennani, Hybrid Peer-to-Peer Model in Proximity Applications, Int. Conf. on Advanced Information Networking and Applications (AINA), 2003.
17. M. Thilliez, T. Delot, A Localization Service for Proximity Applications, Workshop on Mobile and Ubiquitous Information Access in conjunction with the 5th Int. Symposium on Human Computer Interaction with Mobile Devices and Services, 2003.
18. M. Thilliez, Y. Colmant, T. Delot, Query Evaluation in ISLANDS, Les 19èmes journées de Bases de données Avancées (BDA), 2003.
19. U. Varshney, R. Vetter, Emerging Mobile and Wireless Networks, Communications of the ACM, Vol. 43, Issue 6, 2000.
20. O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang, Moving objects databases: Issues and solutions, Int. Conf. on Scientific and Statistical Database Management (SSDBM), 1998.
21. O. Wolfson, A. P. Sistla , B. Xu, J. Zhou, S. Chamberlain, DOMINO: Databases fOr MovINg Objects tracking, Int. SIGMOD Conference, 1999.
22. O. Wolfson, A. P. Sistla, S. Chamberlain, Y. Yesha, Updating and Querying Databases that Track Mobile Units, Distributed and Parallel Databases Journal on Mobile Data Management and Applications, 1999.
23. O. Wolfson, S. Chamberlain, K. Kalpakis, Y. Yesha, Modeling Moving Objects for Location Based Services, NSF Workshop on Infrastructure for Mobile and Wireless Systems, 2001.
24. B. Yang, H. Garcia-Molina, Comparing Hybrid Peer-To-Peer System, Int. Conf. on Very Large DataBases (VLDB), 2001.
25. J. Zhang, L. Gruenwald, Prioritized Sequencing for efficient Query on Broadcast Geographical Information on Mobile-Computing, ACM Int. Symp. on Advances in geographic information systems, (GIS), 2002.